Managing Two-Factor Authentication Setup Through Password Managers

Jonathan William Dutson
Brigham Young University

Follow this and additional works at: https://scholarsarchive.byu.edu/etd

Part of the Physical Sciences and Mathematics Commons

BYU ScholarsArchive Citation
https://scholarsarchive.byu.edu/etd/8976

This Thesis is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact ellen_amatangelo@byu.edu.
Managing Two-Factor Authentication Setup
Through Password Managers

Jonathan W. Dutson

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of
Master of Science

Kent Seamons, Chair
Daniel Zappala
Jacob Crandall

Department of Computer Science
Brigham Young University

Copyright © 2020 Jonathan W. Dutson
All Rights Reserved
ABSTRACT

Managing Two-Factor Authentication Setup Through Password Managers

Jonathan W. Dutson
Department of Computer Science, BYU
Master of Science

Two-factor authentication (2FA) provides online accounts with protection against remote account compromise. Despite the security benefits, adoption of 2FA has remained low, in part due to poor usability. We explore the possibility of improving the usability of the 2FA setup process by providing setup automation through password managers. We create a proof-of-concept KeePass (a popular password manager) extension that adds browser-based automation to the 2FA setup process and conduct a 30-participant within-subjects user study to measure user perceptions about the system. Our system is found to be significantly more usable than the current manual method of 2FA setup for multiple online accounts, with our system receiving an average SUS score of ‘A’ while the manual setup method received an average score of ‘D’. We conduct a meta-analysis of some of the most common methods of 2FA used by websites today and propose a web API that could increase the speed, ease, and scalability of 2FA setup automation. Our threat analysis suggests that using password managers for 2FA automation can be implemented without introducing significant security risks to the process. The promising results from our user study and analysis indicate that password managers have strong potential for improving the usability of 2FA setup.

Keywords: two-factor authentication, 2FA, password managers, security, usability
ACKNOWLEDGMENTS

I thank my parents for their constant love and encouragement throughout my years as a university student. I would also like to thank my advisor Kent Seamons for his guidance, mentorship, and kindness, as well as to my thesis committee for their support. I’m grateful to Austin Kolander and Clinton McCardell for their help in creating a password manager extension and assistance with conducting user studies.
# Table of Contents

List of Figures vii

List of Tables viii

List of Listings ix

## I Introduction 1

1 Introduction 2

2 Related Work 6

2.1 2FA .................................................. 6

2.2 Password Managers .................................. 8

## II System Design 11

3 Background 12

3.1 Common Second-Factor Methods ....................... 12

3.1.1 Backup Codes ...................................... 12

3.1.2 Push Notifications .................................. 12

3.1.3 Phone Calling ...................................... 13

3.1.4 SMS ................................................ 13

3.1.5 TOTP ............................................... 13

3.1.6 U2F ................................................ 14
### List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>API endpoints</td>
<td>20</td>
</tr>
<tr>
<td>4.2</td>
<td>Flow of 2FA setup with proposed API</td>
<td>20</td>
</tr>
<tr>
<td>4.3</td>
<td>KeePass user interface</td>
<td>24</td>
</tr>
<tr>
<td>4.4</td>
<td>Navigating to 2FA setup extension menu</td>
<td>25</td>
</tr>
<tr>
<td>4.5</td>
<td>KeePass 2FA setup extension menu</td>
<td>25</td>
</tr>
<tr>
<td>4.6</td>
<td>Selection of 2FA method for Reddit (this example shows only one option because Reddit only offers app-based TOTP 2FA)</td>
<td>26</td>
</tr>
<tr>
<td>4.7</td>
<td>Loading screen for Reddit 2FA setup</td>
<td>26</td>
</tr>
<tr>
<td>4.8</td>
<td>QR code and instructions during Reddit 2FA setup</td>
<td>27</td>
</tr>
<tr>
<td>4.9</td>
<td>Success message after Reddit 2FA setup</td>
<td>27</td>
</tr>
<tr>
<td>5.1</td>
<td>Password manager use in study</td>
<td>34</td>
</tr>
<tr>
<td>6.1</td>
<td>Time distribution of 2FA setup</td>
<td>36</td>
</tr>
<tr>
<td>6.2</td>
<td>Score distribution and mean for ASQ Q4: Ease of completing the task</td>
<td>37</td>
</tr>
<tr>
<td>6.3</td>
<td>Twitter’s settings menu</td>
<td>38</td>
</tr>
<tr>
<td>6.4</td>
<td>Score distribution and mean for ASQ Q4: Amount of time to complete the task</td>
<td>39</td>
</tr>
<tr>
<td>6.5</td>
<td>Score distribution and mean for ASQ Q3: Support information when completing the task</td>
<td>40</td>
</tr>
<tr>
<td>6.6</td>
<td>SUS score distribution</td>
<td>41</td>
</tr>
<tr>
<td>A.1</td>
<td>Flyers used to recruit participants for within-subjects user study</td>
<td>59</td>
</tr>
</tbody>
</table>
List of Tables

4.1 2FA method with corresponding identifier and response to the website’s challenge. 19
4.2 2FA method with corresponding identifier, direction identifier is shared, and identifier transfer object. 21
5.1 The five websites tested with corresponding second factors 29
6.1 Pearson correlation coefficient and p-values from comparison of Google ASQ scores between users who experienced technical difficulties during the manual setup of 2FA for Google. 44
List of Listings

swagger.yaml ................................................................. 60
Part I

Introduction
Two-factor authentication (2FA) and password managers are two security approaches that can help prevent account compromise. 2FA requires users to authenticate with two out of three authentication factors:

1. Something you know
2. Something you have
3. Something you are

The vast majority of second factors used today are in the “something you have” category. 2FA protects against account compromise through credential theft alone and can provide resistance to other attacks including man-in-the-middle (MITM) and phishing attacks. Accounts without 2FA enabled are much more likely to be compromised. For example, at RSA 2020 Microsoft engineers revealed that 99.9% of tracked compromised accounts did not use 2FA, with over 1.2 million accounts compromised in January 2020 [35].

Most 2FA methods involve entering a code from a second factor in addition to a username and password. These codes are obtained through various means, such as text messages, push notifications, or hardware code generators. Other second-factor methods include push notifications, Universal Second Factor (U2F) tokens, and printing backup codes to be used later.

While 2FA provides increases account security, it comes at a cost of usability. One barrier to 2FA adoption is the usability of the setup process. The day-to-day use of 2FA
has been found to be more usable than the process of setting up 2FA [31], suggesting that overcoming the setup hurdle is a critical step in increasing 2FA adoption.

2FA setup is especially difficult to do on a large scale. There are a wide variety of 2FA methods, and each has a unique setup process, some being more usable than others [30]. In addition to handling a variety of 2FA methods, users have to manage variations between each website they want to protect with 2FA; hundreds of websites offer 2FA [10], but each has its own setup interface, workflow, and terminology. This makes setting up or re-establishing 2FA over multiple accounts a time-consuming process. A user who replaces their device (e.g., getting a new cell phone) may have to re-establish 2FA across many accounts en masse, leading to a poor usability experience, as renowned security researcher Matt Blaze recently tweeted: “Upgraded to a new phone, which is like taking a 12 hour refresher course in configuring 2FA” [6]. We are not aware of any previous research that has addressed the usability challenges of setting up 2FA on multiple accounts.

Password managers may improve both the usability and security of a user’s online presence [19]. It is unreasonable to expect users to remember strong and unique passwords for each one of their online accounts. Because this is difficult, users often choose weak passwords and reuse passwords across multiple websites. Password managers address the problems of weak passwords by providing secure password generation. They also take the burden of remembering passwords off of the user, with most password managers requiring users to remember only the password to the manager itself: a “master password”.

There are a variety of password managers available for commercial and public use. Most major browsers have a built-in password manager. While this can be convenient for many users (there is no additional download or installation needed beyond the browser), many security experts recommend using dedicated password managers [36]. Browser-based managers are much better than using weak passwords, but they have unique security flaws [37] that may not be present in dedicated password managers. Some dedicated password managers store data only locally, while others offer sync and backup through the cloud. Local
data storage prevents attackers from stealing user data from a server. Many users opt for the convenience of cloud storage. While this requires trusting a third-party to securely store the user’s data, it allows users to sync passwords between devices or access passwords online.

Like 2FA, the wide-scale adoption of password managers has been difficult. However, while 2FA decreases the perceived usability of user authentication, using a password manager may increases usability by eliminating the burden of remembering multiple usernames and passwords. Some users have developed their own method of storing or remembering passwords, such as keeping passwords written down, stored in a document on their computer (with or without some sort of security measure), or reusing passwords (or slightly modified passwords) across many sites [21].

Our research is the first to explore how password managers may be used to improve the user experience of 2FA setup. Much of the previous research on 2FA usability has focused on the usability of individual 2FA methods or individual websites. In contrast, our work focuses on improving the 2FA setup process irrespective of method or website. Our main contributions include:

1. Providing the first analysis of potential benefits and risks of using password managers to manage 2FA setup.
2. Developing a 2FA setup automation system through a KeePass (a popular password manager) extension.
3. Creating a standardized web API for 2FA setup.
4. Conducting a user study to compare the usability of 2FA setup through a password manager with manual 2FA setup.

In the same way that password managers may be used to improve the usability of passwords over multiple websites, we demonstrate that a 2FA manager may be used to improve the usability and scalability of 2FA adoption. Providing users with a single point for
managing 2FA across multiple online accounts will make the experience of setting up and changing 2FA much more usable.

Our research has the potential to benefit many people and institutions. 2FA management through password managers allows users to more easily register or change their 2FA methods or hardware (such as when a user gets a new phone or U2F device). 2FA is one of the most effective ways to mitigate threats from remote attackers [17, 18]. Users do not want their accounts accessed by unauthorized intruders, and improving 2FA setup will enable them to more easily secure their accounts. Online account security is critical to many essential industries, including education, retail, healthcare, technology, and financial services. Using password managers to manage 2FA setup has the potential to improve online account security for companies, governments, and users around the world.
Chapter 2

Related Work

This chapter describes related research on two-factor authentication and password managers.

2.1 2FA

In 2015, Petsas et al. attempted to quantify the adoption of 2FA on Google and concluded that less than 6.4% of Google users have adopted 2FA [29]. Over a three month period, they observed that less than 3% of accounts with 2FA enabled went on to disable the feature. They also found that Google’s password-reminder process can disclose sensitive information (i.e. names and photographs). At Enigma 2018, a software engineer from Google revealed that less than 10% of active Google accounts have 2FA enabled [28]. The low adoption rates for Google accounts are concerning because email accounts are often considered to be critical accounts. If users are unwilling to adopt 2FA for their more important accounts, it is unlikely that they will adopt it for accounts they consider to be less critical.

A few studies have focused on the usability of banking systems in the UK. Gunson et al. studied a portable key fob which outputs a one-time code [16]. They found that while most customers perceived the 2FA method as offering a higher level of security, it was accompanied by significantly lower perceptions about usability and convenience. The participants studied by Kol et al. were generally dissatisfied with the experience of using hardware tokens to authenticate, with one participant switching banks to avoid using the second-factor [24]. They recommend reducing the number of steps required for authentication
to as few as possible to provide a more usable authentication experience. Our proposal builds off of this recommendation by attempting to reduce the number of steps needed to set up 2FA on multiple websites.

In 2015 Karapanos et al. proposed a new 2FA system called Sound-Proof. [22] This system records ambient noise using the microphones on a user’s phone and on the device the user is attempting to authenticate with. It then compares the two audio recordings to determine whether the devices are in the same proximity. Their system requires no user interaction, and so the user experience is similar to authentication with username and password only. Users ranked Sound-Proof to be more usable than Google’s two-factor authentication.

Reynolds et al. separated the study of the YubiKey’s day-to-day usability with the usability of the setup process [31]. Separating the setup and day-to-day usage of the device helped mitigate bias that may occur when a negative or positive setup experience is reflected onto the day-to-day usability or vice versa. They found that while most users were pleased with the day-to-day usability of the YubiKey, the setup process was viewed poorly. They recommend that the 2FA setup process be standardized to improve usability. Reese et al. similarly separated the study of day-to-day and setup usability [30]. They studied five second-factors and concluded that well-implemented 2FA could generally be usable in both setup and daily use. However, they acknowledge that the university population they studied is not representative of the general population, and recommend that other populations (such as the elderly or those without a college education) be studied.

De Cristofaro et al. compared the usability of three second-factors and found that perceptions of 2FA were more correlated with an individual’s background rather than the specific technology [11]. Most users studied found each of the methods studied to be highly usable. They find security tokens to be the most common second factor used in work environments, while email or SMS codes are the most often used in personal and financial environments. Colnago et al. studied the adoption of Duo, a cloud 2FA provider, at a
university [9]. They found that users who have not tried 2FA anticipate it being more inconvenient and more difficult to use than it actually is. Once users tried 2FA, many thought it was easier to use than expected. They suggest that organizations should require the use of 2FA, with the understanding that once users begin to use 2FA they may have fewer usability concerns.

This previous work has highlighted many usability problems with 2FA and has provided suggestions for improving its user experience. While previous work has suggested standardizing the 2FA registration process, no work has developed such a system. As far as we are aware, no previous work has explored the possibility of automating 2FA setup. Our work provides a first look into how the usability of 2FA may be increased by removing many of the steps required for the setup of the second factor.

2.2 Password Managers

In 2006 Chiasson and van Oorschot studied two password managers and found that the most significant problems came from inaccurate mental models of the tools [8]. Many users expressed reluctance about the idea of voluntary adoption of password managers, citing reasons such as discomfort with giving control of passwords to the manager or feeling like using a password manager was unnecessary. Fagan et al. collected opinions about password managers from users and non-users [13]. They found that users of password managers generally cited convenience and usefulness as their motivation for using a password manager rather than security concerns, while non-users mentioned security concerns as the primary reason for not using a password manager. Alkaldi and Renaud found that lack of trust and lack of awareness about smartphone password managers prevented many users from adopting them [4]. They suggest that smartphone password manager designers should focus more on the user experience.

Karole et al. compared the usability of three kinds of password managers: an online manager, a phone manager, and a USB manager [23]. They found that while the online
manager had better usability, users preferred the two portable managers. They attributed this to users being uncomfortable giving their passwords to an online program.

Password managers are subject to a variety of security risks. Multiple papers have discovered exploits that can take advantage of password managers that auto-fill password forms [1, 14, 33]. Attackers can create a malicious invisible form that tricks a password manager into entering in a victim’s credentials, leaking them to the attacker. While password managers have taken measures against these exploits (such as comparing the website URL with the URL of the credential being auto-filled), some still recommend disabling auto-fill capabilities. Li et al. analyze five online password managers and identify four classes of vulnerabilities [26]. They find that attackers could steal arbitrary credentials from four of the five tools studied. The discovered vulnerabilities spanned a wide range of features, including one-time passwords, bookmarklets, and shared passwords. Belenko and Sklyarov analyzed over twelve smartphone password manager applications and found that many had inadequate cryptographic protection or misuse protection offered by the OS [5]. They find the protection offered by the device OS (Apple and BlackBerry in their study) to be more secure than the analyzed applications. Zhao et al. describe several decryption, decoding, and brute force attacks [38, 39]. They suggest that master passwords should be analyzed by a password checker to enforce strong passwords and that master passwords should not be associated with data sent to cloud storage. Gray et al. analyze security risks of local password managers and discover that passwords may be found in temp folders, page files, or the recycle bin [15]. Although some of these vulnerabilities are bound to certain conditions (such as a user being logged-in when the attack is launched), the paper notes that these vulnerabilities are still significant, since existing malware targets password managers and could take advantage of these security flaws. Most of the vulnerabilities were found in one of the three password managers studied. The paper suggests that weaker password managers can increase security by adopting security practices from other password managers. Many
managers publish explanations of how they protect user data, making this information readily available for other password managers to study and consider adopting.

Some work has been done to propose improvements to the security and usability of password managers. McCarney proposes a dual-possession password manager that requires two devices (such as a desktop and a phone) in order to operate the password manager [27]. Their system exchanges the security provided by “something you know” (i.e. the master password) with a second “something you have”. Their proof-of-concept implementation encrypts passwords on a desktop and stores them on a smartphone. Stajano et al. propose adding a “password-manager friendly” (PMF) semantic markup to account creation, access, and management forms [34]. This would add classes to form CSS to make it easier for password managers to provide a consistent and usable user experience.
Part II

System Design
Chapter 3

Background

Below we describe some of the 2FA methods most widely-used today. Some methods are more secure than others, and each method has its own usability experience. We propose that password managers may be used to facilitate the setup of each of these methods, along with other methods that may become popular in the future.

3.1 Common Second-Factor Methods

In this section, we cover the following second-factor methods: backup codes, push notifications, phone calling, SMS, TOTP, U2F, and WebAuthn.

3.1.1 Backup Codes

Some sites require users to have at least two methods of authenticating with 2FA. Many websites offer codes as a backup method when the primary method is unavailable (e.g. a 2FA device is lost, out of batteries, etc.). These codes are typically printed out or stored digitally. This prevents users from being locked out of their account even if their primary second factor becomes unavailable.

3.1.2 Push Notifications

Users can enroll their device in a website’s push notifications. When authenticating, websites can send an authentication request through a push notification to an internet-connected device. The user can approve or reject the request with a simple touch. This method is
more secure than SMS or TOTP because the direct device to retailer connection can prevent phishing attacks. However, this method may involve installing an app, which users are sometimes unwilling or reluctant to do.

### 3.1.3 Phone Calling

This method allows users to confirm their identity by answering a phone call. Depending on the implementation, users may receive a code over the phone, or they may simply be required to press a number on their phone to confirm the authentication request. This method is subject to many of the same attacks on SMS-based 2FA.

### 3.1.4 SMS

SMS-based 2FA is one of the most commonly used methods. Users provide websites with a phone number and then confirm ownership of the number by entering a code texted to them by the website. SMS authentication has some important weaknesses. Social engineering may be used to conduct phishing attacks to steal SMS codes sent to the user’s device. Another attack involves using port-out scams to steal a user’s phone number, allowing SMS messages to be routed to an attacker-controlled phone. While SMS is not an ideal form of 2FA, it remains a popular method because of its simplicity and availability.

### 3.1.5 TOTP

Time-based one-time passcodes (TOTP) uses a shared secret for proof of device ownership. The shared secret is created by the website and is transferred to the client device through a QR or hexadecimal code. This is usually done on the client-side by an app (such as Google’s authenticator app). The shared secret is combined with a time-based value to create a cryptographic hash. When authenticating, the user gives this hash to the website. The website compares this hash with the hash it generates and authenticates the user if they
match. This hash is updated regularly, usually every 30 or 60 seconds. Some TOTP providers offer cloud backup, so a user may recover their shared secrets if their device is lost.

3.1.6 U2F

The Universal Second-Factor (U2F) protocol is an authentication standard that allows security keys (such as Yubico’s YubiKey or Google’s Titan) to be used as a second factor. The protocol describes the interactions between a website (relying party), a browser (client), and an authenticator. The only action required from the user is a “proof of presence”, which generally involves pressing a button on the U2F device. Since this user interaction does not require typing a code into a website, U2F provides defense against many of the remote phishing attacks other methods (such as SMS or TOTP) are susceptible to. U2F devices are tamper and clone-resistant. The devices use a private key to generate unique credentials for each registered account. These credentials are encrypted locally and are typically stored with the relying party. This means U2F devices have no personally identifiable information. After adopting U2F, Google virtually eliminated account compromise through phishing attacks [20]. U2F is generally considered one of the most secure second factors widely available today.

3.1.7 WebAuthn

WebAuthn is a standard for authenticating users using public-key cryptography. It is a part of FIDO2, the next evolution of the U2F protocol. In addition to two-factor authentication, the protocol supports passwordless authentication. Passwordless authentication uses credentials stored locally on the WebAuthn authenticator to authenticate without a username and password. This requires an authentication gesture such as a PIN or a biometric. The authenticator can be a roaming hardware authenticator (such as a YubiKey, phone, or smartwatch) or a platform authenticator built into a device (such as Windows Hello). At the time of writing, WebAuthn has been adopted by most major desktop and mobile browsers.
3.2 Existing Password Manager Features

Some popular password managers use automation to help strengthen password security for their users. For example, Dashlane and LastPass can automatically change a user’s password; either through headless browser automation or through an API if one is offered by the website [2, 3]. LastPass supports 75 websites, while Dashlane supports over 300. During this process, Dashlane can detect 2FA requests, and allows the user to authenticate with their second-factor directly from Dashlane.

Automatically changing a user’s password requires various levels of trust from the user. For example, Dashlane’s password changer requires both the old and new passwords to be briefly decrypted on the company’s servers, while automatic password changing through LastPass takes place locally on the user’s machine before being encrypted and synced to LastPass servers. This means that Dashlane users who use this feature must be comfortable with trusting Dashlane with access to their plaintext password.

LastPass and 1Password provide TOTP-compliant authenticator apps and offer cloud backup for TOTP secrets. However, some users are uncomfortable with the idea of relying on a password manager for 2FA secret storage, since this creates a single point of failure (i.e. if a hacker compromises the password manager, the benefit of 2FA is nullified). Our proposed system does not create a single point of failure since the password manager is not used as a second factor (or storage for a 2FA key) but instead is used only to facilitate the 2FA setup process.
Chapter 4

2FA Setup Management through Password Managers

To improve the usability and scalability of 2FA adoption, we propose a system for partially automating the 2FA setup process. We consider password managers to be an ideal platform for implementing our proposed automation. Password managers are already considered to be trustworthy security tools, and users may be more likely to adopt a new feature to an existing trusted system than they would be to adopt an entirely new system. Adding a second factor to a website requires user authentication, and since password managers already store user credentials, they can reduce the amount of user interaction needed for authentication.

Password managers might offer additional features to further improve the management of 2FA setup. For example, a password manager could compare a user’s accounts with published lists of websites that offer 2FA and prompt the user to set up 2FA, potentially increasing the number of websites for which a user may adopt 2FA. Password managers could also offer to automatically store backup codes for the user, offering increased convenience in exchange for some new security risks.

4.1 Design

We identify two ways through which password managers could provide 2FA setup automation: (1) web automation that mimics a user setting up 2FA, and (2) a standardized API provided by the website. We compare these two methods below.
4.1.1 Web Automation

Web automation is used for a variety of tasks such as web performance testing and data extraction. A password manager could use automation scripts to visit a website, authenticate the user, and programmatically navigate the site to request 2FA setup. This automation can reduce the number of steps a user needs to take to set up 2FA, providing a more usable experience [24]. The major advantage of web automation is that it allows password managers to provide 2FA setup automation without cooperation from websites. However, the differences in user interfaces and 2FA setup processes between websites make it necessary to implement a specialized web automation script for each supported website.

In order to set up 2FA, the password manager first needs to authenticate the user. Web automation should be able to authenticate a user whether or not they have already enabled 2FA. If 2FA is not enabled on the website, the password manager can authenticate the user through web automation alone. If 2FA is enabled, the web automation should recognize which methods are enabled and may either prompt the user to select which method of 2FA they would like to authenticate with or use a default 2FA method. The user should then be prompted to take action based on the chosen 2FA method.

Because of the need to tailor automation to each website, web automation is neither a scalable nor a robust option for 2FA setup management. Each website for which the password manager supports 2FA automation requires a unique script, making it difficult to scale to a large number of websites. Web automation scripts are dependant on a website’s underlying HTML, so changes to a website’s flow, content, or design may break existing automation. Since the scripts are not robust enough to independently adjust to changes to a website’s interface, maintaining support for a website will require consistent monitoring and updating of the automation scripts to detect errors and restore functionality.
4.1.2 2FA Setup API

While browser automation would allow modern password managers to automate 2FA on virtually any website, relying on browser automation is neither scalable nor robust, as discussed in the previous section. In this section, we propose a standardized API for setting up 2FA which websites could adopt to simplify the setup process. Implementation of our API would allow password managers to support 2FA setup automation for many websites without the creation of unique automation scripts, making this a more scalable and robust option for automated 2FA setup.

As a first step towards developing a standardized API, we performed a meta-analysis of some of the most common 2FA methods in use today to determine how they would fit into a 2FA setup API. Each of the most common second factors used in online authentication today is in the “something you have” category. Proof of ownership of the “something you have” is obtained by sharing a unique identifier between the user and the website. These identifiers are either provided to the website by the user or provided to the user by the website and are used to complete challenge-response authentication.

Technically, all that is needed in order to setup 2FA is the sharing of the identifier between user and website. In the simplest case of backup codes, the identifier is a one-time use code that is transferred from the website to the user. Once the identifier is sent to the user, setup is considered complete. However, most forms of 2FA are more complex than backup codes. Websites want to verify that the user has possession of the identifier. The identifier for SMS messages is a phone number transferred from the user to the website. While it would be possible for a website to enable SMS as a second factor without verification of a user’s phone number, this would cause users to be unable to authenticate if they had entered in their phone number incorrectly, potentially locking the user out of their account.

To prevent account lockout, websites require users to prove possession of the identifier before enabling 2FA. This proof-of-possession occurs through challenge-response authentication. During challenge-response authentication, the website issues a challenge (a question
<table>
<thead>
<tr>
<th>Method</th>
<th>Identifier</th>
<th>Challenge Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backup Codes</td>
<td>Backup code</td>
<td>Backup code</td>
</tr>
<tr>
<td>Phone Calling</td>
<td>Phone number</td>
<td>Code received through call</td>
</tr>
<tr>
<td>Push Notification</td>
<td>Registration ID</td>
<td>Approval of authentication request</td>
</tr>
<tr>
<td>SMS</td>
<td>Phone number</td>
<td>Code received through text</td>
</tr>
<tr>
<td>TOTP</td>
<td>Shared secret</td>
<td>Code generated from shared secret and timestamp</td>
</tr>
<tr>
<td>U2F</td>
<td>Public key</td>
<td>Signed challenge</td>
</tr>
<tr>
<td>WebAuthn</td>
<td>Public key</td>
<td>Signed challenge</td>
</tr>
</tbody>
</table>

Table 4.1: 2FA method with corresponding identifier and response to the website’s challenge.

for which the answer requires possession of the identifier) to the user, who must answer the question with the proper response (proof of possession of the identifier) in order to authenticate. Examples of identifiers and challenges for common second factors are listed in Table 4.1.

After performing a meta-analysis of some of the most commonly used 2FA methods, we developed a proof-of-concept RESTful Web API based on challenge-response authentication. Our API is designed to be relatively simple to implement but robust enough to support a wide variety of second factors. Authenticated users or password managers (referred to as the client below) can use five endpoints (see Figure 4.1) to manage their 2FA methods. 2FA setup occurs through the exchange of three objects between the client and website:

1. Request object
2. Challenge object
3. Response Object

This process is visualized in Figure 4.2. During this exchange, the method’s identifier is shared between the client and website. Table 4.2 shows the direction the identifier is shared (i.e. whether the client provides the identifier to the website or vice versa) and in which object the identifier is transferred.

The website issues a challenge to the user and enables the 2FA method on successful completion of the challenge by the client. The three objects are method-specific, and some of
the objects may be empty for some methods. For example, for SMS verification, the request object contains the phone number the client wants to receive SMS verification messages. The challenge object may be empty since the challenge is sent out-of-band through the phone network. On receipt of the response object (containing the SMS verification code received by the client) the website can enable SMS verification. YAML (a human-friendly data-serialization language) code for this API and example objects for SMS, TOTP, backup codes, and U2F verification can be seen in appendix A.4.

For 2FA methods such as U2F or WebAuthn, the password manager must implement the Client to Authenticator Protocol (CTAP) in order to communicate with the authenticator.
Table 4.2: 2FA method with corresponding identifier, direction identifier is shared, and identifier transfer object.

(i.e. the hardware token). The 2FA setup API then acts essentially the same as the U2F or WebAuthn protocols.

Just as with the browser-automation method, if a website already has 2FA enabled then the password manager will need to authenticate with 2FA before setting up another second factor or changing the users 2FA settings. However, the authentication process is nearly identical to the setup process. The authentication process uses the same challenge-response verification, but does not require the sharing of the identifier between the user and the website, since this identifier was already shared during the setup process. This is what will allow our proposed API to be used for both setup and authentication before the setup process. For example, if a user wants to add TOTP as a second factor to a website that supports the setup API, but already has SMS enabled as a second factor, the password manager will be able to use the API to authenticate with SMS using the same challenge-response pattern used in SMS setup.

4.1.3 Threat Analysis

Since our 2FA management system relies on password managers, it is subject to the same security risks and vulnerabilities of the password manager our system is built on. It is important to determine whether our proposed system introduces or exacerbates security risks during user authentication. Our threat analysis assumes the following:
1. The password manager communicates with websites only over HTTPS and TLS.

2. Network attacks such as eavesdroppers, man-in-the-middle attacks, are out of scope. These attacks target network protocols rather than our system design.

3. The website is trustworthy and has not been compromised.

4. The device and application on which the user runs our system are trustworthy have not been compromised.

5. The password manager assists with 2FA setup but does not store any 2FA identifiers

The most significant risk introduced by our system is when a user’s password manager is compromised. If this occurs, an attacker could use our system to more easily set up 2FA on multiple websites using a second factor controlled by the attacker, locking the legitimate user out of their accounts. Of course, attackers with access to a user’s password manager can already manually set up 2FA on compromised accounts, but our system would allow an attacker to do this more scalably. An improvement in the usability of the system for the user also makes it more usable for attackers who compromise the system. However, for user accounts where the user has already enabled 2FA, an attacker who has compromised the user’s password manager will be prevented from compromising the account unless they are able to obtain the user’s second factor.

Users may choose to sacrifice some security for improved usability by breaking our last assumption. Some websites (such as Github) recommend storing backup 2FA codes in a password manager. This process could be simplified using our proposed automation system; backup codes could automatically be stored on the password manager. In this case, if an attacker compromises a user’s password manager, they would be able to bypass 2FA for any websites where the backup codes are stored with the password manager. Deciding whether to store backup codes on a password manager requires considering the trade-off between usability and security.
The risk of password manager compromise could be mitigated by strongly encouraging (or requiring) 2FA to be enabled on the password manager. If 2FA is enabled for a password manager, then data stored on the password manager (including backup codes) would still be protected by a second factor. Whether or not a user stores backup codes in their password manager, we strongly recommend that users protect their password manager with 2FA.

Password managers can provide strong protection against both online and offline attacks through secure password generation. However, password manager users may still reuse passwords or use weak passwords. Offline attacks may compromise moderately strong passwords if they are not properly salted and hashed. Even the strongest passwords can be stolen if they are stored unencrypted. 2FA protects against account compromise even after credential theft. For this reason, using a password manager in combination with 2FA provides strong protection from account compromise. We judge the small risks associated with our system to be worth the potential benefits for the usability of 2FA setup.

4.2 Implementation

We created an extension for the popular open-source password manager KeePass (see Figure 4.3) to implement browser-based 2FA setup automation. The code for our extension has been published at https://bitbucket.org/isrlauth/keepass-2fa-setup/.

The first step in the setup process is to select a website where a new 2FA method will be set up. The user first clicks on the “Tools” menu in the KeePass application, and selects the “Two Factor Setup...” option, as seen in Figure 4.4. This opens a new window with a menu (see Figure 4.5) from which a user can set up 2FA from supported websites. The menu shows all supported websites for which the user has credentials stored in KeePass. After selecting a website, the user can select one of the 2FA methods supported by the website (see Figure 4.6). The user enters needed information (i.e. phone number) and the extension then launches a headless browser to perform 2FA setup.
The extension shows a loading screen (see Figure 4.7) until setup is complete or user interaction is needed. For example, when enrolling in SMS 2FA, the extension prompts the user for the code after the website sends a text to the provided phone number. For TOTP-based 2FA, the extension prompts the user to scan a QR code with an authenticator app and then enter the code provided from the app. Figure 4.8 shows this stage of TOTP setup for Reddit. On successful completion of the second factor setup, the extension notifies the user (see Figure 4.9), who can then return to the 2FA management menu. The instructional text and success messages were copied from the supported websites, with occasional superficial adjustments.

We designed our password manager extension with the intention of using it in a user study about using password managers to set up 2FA. The next two chapters describe the design and results of our user study.
Figure 4.4: Navigating to 2FA setup extension menu

Figure 4.5: KeePass 2FA setup extension menu

Select a website to begin two-factor authentication setup

Facebook
Reddit
Google
Pinterest
Amazon
GitHub
Twitter
Figure 4.6: Selection of 2FA method for Reddit (this example shows only one option because Reddit only offers app-based TOTP 2FA)

Figure 4.7: Loading screen for Reddit 2FA setup
1. Access an authenticator app like Google Authenticator or Authy on your mobile device. Select an option to setup a new account and follow the instructions given to you in your app.

2. Use the app to scan the barcode above or enter the key manually.

   TWIKPQM3CER15HF6NHPCH4YS6LQMSGBA3

3. Enter the verification code generated by your authenticator app to complete setup.

   940523

---

Figure 4.9: Success message after Reddit 2FA setup

Success! You have enabled two-factor authentication.

You will now need to enter the verification code from your authenticator app when you login to Reddit on desktop or mobile. You can disable this at any time in your preferences.

Be sure to generate your backup codes! This will allow you access to your account in the event you lose your mobile device.

Make sure your Reddit iOS / Android app is updated to the most recent version to ensure two-factor authentication works properly.
Part III

User Study
Chapter 5

Methodology

We conducted a 30-person within-subjects laboratory study approved by our institution’s ethics review board to evaluate the usability of our proposed system.

5.1 Study Design

We recruited 30 participants to set up 2FA on five websites (see Table 5.1) both manually and using the password manager extension we built to partially-automate the setup process. We calculated our sample size using Sauro and Lewis’s instructions for estimating sample size for a within-subjects comparison of an alternative [32]. The participants were evenly split into two groups: A and B. In group A, users first set up 2FA on the five websites manually. After completing all five websites, they then set up 2FA using our automated system. The ordering was switched for users in group B, who set up 2FA using our system before setting up 2FA for the websites manually. We will refer to the process of manually setting up 2FA on five websites as M-2FA, and the automated process as A-2FA.

<table>
<thead>
<tr>
<th>Website</th>
<th>Second Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>GitHub</td>
<td>TOTP</td>
</tr>
<tr>
<td>Google</td>
<td>Google Prompt</td>
</tr>
<tr>
<td>Pinterest</td>
<td>SMS</td>
</tr>
<tr>
<td>Reddit</td>
<td>TOTP</td>
</tr>
<tr>
<td>Twitter</td>
<td>SMS</td>
</tr>
</tbody>
</table>

Table 5.1: The five websites tested with corresponding second factors
After completing both M-2FA and A-2FA, users completed two SUS questionnaires, one for each approach. They answered a few additional questions to measure their openness to using A-2FA. Users authenticated to test accounts on each of the websites to eliminate the risk of exposing the user’s personal account information. We created separate test accounts for A-2FA and M-2FA since if the processes shared accounts we would have had to spend time disabling 2FA from the test accounts in the middle of each user study. The test credentials were stored in two KeePass password managers on two virtual machines, one with the credentials for the automated process, and the other with the credentials for the manual process. This removed the need for the participant to have any knowledge about account usernames or passwords. However, our extension was only installed on the virtual machine used during the automated process. Screen recordings captured the participant’s activity on the computer and were analyzed for timing data.

To counterbalance sequential effects from the order of the websites during setup, we generated 10 different orderings using two five-by-five balanced Latin squares [25]. Each participant used the same ordering during the manual and automated process, and each ordering was used 3 times.

5.1.1 The After Scenario Questionnaire

We used the After Scenario Questionnaire (ASQ) to measure participant sentiment about the usability of the setup process for individual websites [32]. The ASQ is a standard usability questionnaire composed of three questions:

1. Overall, I am satisfied with the ease of completing the tasks in this scenario

2. Overall, I am satisfied with the amount of time it took to complete the tasks in this scenario

3. Overall, I am satisfied with the support information (online help, messages, documentation) when completing the tasks
Participants rate each of the above questions on a Likert scale of 1 to 7, with 1 being “Strongly agree” and 7 being “Strongly disagree”.

### 5.1.2 The System Usability Scale

We used the System Usability Scale (SUS) to measure participant sentiment about the usability of the A-2FA and M-2FA processes as a whole (i.e. not the usability of specific websites, but rather the usability of setting up 2FA across multiple websites at one time). The System Usability Scale is composed of 10 Likert scale questions, with 1 being “Strongly disagree” and 5 being “Strongly agree” [7].

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

The responses to these questions are used to calculate a SUS score from 0 to 100. An analysis of over 5000 users across 446 studies found that the average SUS score is 68 with a standard deviation of 12.5 [32]. These scores are often interpreted using percentile rankings or by assigning SUS scores a letter grade.
5.1.3 M-2FA

Participants opened a Chrome browser and were told how to fill in credentials for each website using KeePass. There were multiple ways for a participant to fill credentials, including a KeePass Chrome extension or copying and pasting usernames and passwords from the KeePass application. The study coordinator explained to the participant how to use the KeePass Chrome extension and instructed the participant to set up 2FA on the five websites, using any online resources (such as Google or the website’s provided documentation) as needed. Participants were told they did not have to save or keep track of backup codes, which are a part of the setup process for some of the studied websites. After completing each website, participants answered the ASQ. Participants were instructed to use the ASQ to rate their setup experience specific to the website on which they were enabling 2FA.

5.1.4 A-2FA

The study coordinator showed the participant where to find the 2FA setup menu from the automated 2FA setup KeePass extension and asked them to use the extension to set up 2FA on the five websites. Participants answered the ASQ after each performing setup for each website.

5.1.5 Exit survey

After completing both M-2FA and A-2FA, users completed an online survey asking demographic questions as well as questions about the setup experience. Participants completed a SUS questionnaire for both processes and were asked to provide feedback on the positive and negative aspects of each process. The SUS questionnaire was used to measure the usability of M-2FA and A-2FA setup as a whole rather than the usability of individual websites.
5.2 Recruitment

We recruited 30 participants who were familiar with two-factor authentication and who used a dedicated password manager. Users who only used a browser-based password manager such as the Chrome or Firefox password managers were not eligible for the study since we hypothesized that individuals who used a dedicated password manager were more likely to have intentionally decided to use a password manager and would be more aware of additional features offered by their password manager. Participants were recruited using flyers posted across the campus of Brigham Young University (BYU) and through a recruitment message distributed through email and Slack channels. After signing up for the study, participants were randomly assigned to one of the two study groups. Technical difficulties during three of the studies prompted us to exclude them from analysis. We recruited an additional three participants to repeat the studies, bringing our total participant recruitment to 33. However, only the 30 valid studies were used for analyzing our results. After all user studies had concluded, we noticed one strong outlier in our SUS data. We hypothesized that this was because the standard ASQ and SUS questionnaire used opposite Likert scales (with 1 being “Strongly agree” in the ASQ, and being “Strongly disagree” in the SUS questionnaire). We met with the participant, who confirmed that they had misread the Likert scale when filling out the SUS questionnaires. We exclude this participant’s SUS scores during analysis.

5.3 Demographics

Our participants were young and highly educated, with 27 (90%) ages 18-24 and 3 (10%) ages 25-34. 23 (77%) had some college experience, 1 (3%) had an associate degree, 4 (13%) had a bachelor’s degree, and 2 (7%) had a master’s degree. All participants considered themselves to be average or above in their computer savviness. The distribution of password managers used can be seen in figure 5.1. The most common password manager for participants in the
study was iCloud Keychain, with over a third of participants using the password manager. Lastpass was the second most common, followed by KeePass.

5.4 Compensation

For their time (approximately 45 minutes), participants were compensated with 15 USD at the conclusion of the study.
Chapter 6

Results

In this chapter, we analyze and discuss results from quantitative and qualitative data gathered during our user study.

6.1 Timing

A researcher reviewed each screen recording to gather timing data for each website and each process. For M-2FA, the time began once the user navigated to the website and ended once 2FA setup was complete. For A-2FA, the time began once the user had selected a website from the 2FA extension menu, and ended once setup was complete. In both cases, setup was considered complete after the second factor had been set up and the user had finished reading any information provided by the website after a successful setup. For each of the five websites, the average setup was faster using A-2FA (see Figure 6.1). During M-2FA, Google was the most time-consuming. This is explained in part by the fact that Google was the only method that has users set up a backup second factor. It is also influenced by the multiple technical issues experienced during the manual Google Prompt setup. Twitter was the second slowest during the M-2FA, but the fastest during A-2FA.

6.2 Usability

We measured the usability of individual websites using the ASQ and measured the usability of M-2FA and A-2FA using the SUS. For the ASQ, lower scores represent greater usability. The opposite is true for SUS scores.
6.2.1 ASQ scores

Average participant perceptions about the ease of completing 2FA setup can be seen in Figure 6.2. For every website, participants generally rated A-2FA as being easier to complete than M-2FA. The difference is most prominent with Twitter and least prominent with Reddit. Many users had difficulty finding security settings when navigating Twitter's website. Unlike the other websites, Twitter's setting navigation bar does not list security as one of the main options. Security is instead listed under the account options, as can be seen in Figure 6.3. This caused some participants to spend much longer trying to find the security settings on Twitter compared to the other websites.

Average participant satisfaction about the time to complete 2FA setup on a website can be seen in Figure 6.4. The difference between A-2FA and M-2FA was mixed, with GitHub, Google, and Twitter being rated as better during A-2FA, and Pinterest and Reddit being rated as better during M-2FA. This is interesting given the fact that the average time to complete the task was faster for every website during A-2FA. While the average time may
have been faster during A-2FA, participants were actively interacting with the website during M-2FA, while A-2FA involved a significant amount of waiting for the password manager tool to initiate and complete the setup process. Dellaert and Kahn found that waiting can have a significant effect on user evaluations of websites [12]. In our study, waiting times likely led to decreased satisfaction when using A-2FA. Changing the implementation of the password manager extension could mitigate some of these negative effects. For example, Dellaert and Kahn find that removing uncertainty about the length of the waiting period can minimize the negative effects of waiting. We suggest that future 2FA setup automation include countdowns or duration time information to minimize these effects.
Like the question about ease of task completion, average satisfaction with provided support information was greater for A-2FA than for M-2FA across all five websites, as seen in Figure 6.5. The support information used most often was the instructional text provided from the website during M-2FA. The instructional text from the website was used word-for-word during A-2FA, although it was provided in pop-up boxes rather than on the webpage. While some participants used Google to look for support information during M-2FA, most did not. Since the instructional text was the same during both processes, we hypothesize that the difference in support information satisfaction between A-2FA and M-2FA is primarily a reflection of user perceptions about the usability of the system. If this is true, it shows that the 3 questions that compose the ASQ are not scored independently.

### 6.2.2 SUS scores

Given our sample size of 29 users (the SUS score was dropped for one user who had incorrectly read the Likert scale), we are able to detect a SUS score difference of 5 with 95% confidence,
Figure 6.4: Score distribution and mean for ASQ Q4: Amount of time to complete the task assuming an estimated variance of 156. This variance was calculated using data from about 500 studies, which found that the SUS score had a standard deviation of 12.5 [32]. Users completed the SUS questionnaire for both the automated and manual systems. They were instructed to rate the setup process as a whole, rather than the experience of setting up 2FA on any individual website. We felt that providing users with an opportunity to rate individual websites (through the ASQ) would allow the participants to more accurately express their feelings about the systems as a whole when completing the SUS questionnaires. The distribution of SUS scores can be seen in Figure 6.6.
Figure 6.5: Score distribution and mean for ASQ Q3: Support information when completing the task

The mean SUS score for A-2FA was 82.8 while the mean score for the M-2FA was 60.9. Using the curved grading scale provide by Sauro and Lewis [32], A-2FA received an ‘A’ grade (90-95th percentile) and M-2FA received a grade of ‘D’ (15-34th percentile). The dramatically low SUS score of M-2FA is evidence of the extremely low usability of the current system for setting up 2FA on multiple websites at once. The much higher SUS score for A-2FA is encouraging, suggesting that the 2FA setup process could be made significantly more usable through automation.
6.3 Qualitative

During our exit survey, participants answered open-ended questions about what they liked and disliked about both A-2FA and M-2FA. In this section, we discuss common themes and ideas observed in their responses.

6.3.1 Single Point of Management

Many participants liked that they could set up 2FA from a single user interface when using A-2FA.

P19: “Everything about setting up 2fa was in one location; I didn’t have to navigate a website’s unique layout to find the security page.”
Some participants disliked the inconsistency between websites when trying to set up 2FA during M-2FA.

P29: “Sometimes it was hard to find where setting up the 2FA process was on the website. I didn’t always know exactly where to find it, and sometimes it was in a different place than I thought.”

When asked what they liked about M-2FA, one participant gave an ironic answer:

P26: “It was like a treasure hunt to find things in the UI – so that was ‘fun’ I guess.”

6.3.2 Wait Time

Over half of participants (16; 53%) mentioned the time it took to set up the second factor as something they did not like about A-2FA. Some participants mentioned dislike of the wait time despite having a shorter overall time.

P1: “Even though in the end it probably took less time, no one likes waiting for the thing to load a QR code or send a text, and the system does have a bit of a wait for that.”

Another participant mentioned the lack of information during the wait time as problematic.

P2: “It seemed to take a lot of time and during that time no information was given about what was going on.”

6.4 Discussion

In this section, we discuss additional key findings from our study and address limitations.
6.4.1 Adoptability

Most participants (24; 80%) said they would use A-2FA if a password manager they used offered one. While most participants (23; 77%) trusted the password manager to set up 2FA securely using A-2FA, just under half of the participants (12; 40%) were worried about potential security risks from using a password manager to manage 2FA.

6.4.2 Improving the proof-of-concept system

When participants were asked what they liked least about A-2FA, a few participants mentioned concerns applicable to any implementation of a 2FA automation system. For example, one participant mentioned feeling “sketchy” relying on third-party software to set up 2FA. However, most of their answers were related to our proof-of-concept implementation, not related to the concept of using password managers to provide 2FA management. For example, many users wished that A-2FA was faster. Our proof-of-concept system relied on a series of randomized wait times to avoid anti-automation tools on websites. These wait times could likely be optimized for a faster setup experience. As previously discussed, using an API could dramatically decrease user waiting times. Other feedback was related to problems with the aesthetic or usability of the extension. For example, our extension did not auto-focus on the input text box when a popup appeared prompting the user to enter information. One user described KeePass as “very small and early 2000s-looking”, and suggested that improving the graphics, layout, and fonts of the password manager could do much to improve the user experience. With future iterations of automated 2FA setup software, many of these problems could likely be mitigated, leading to even better usability and adoptability.

6.4.3 Setting up multiple accounts at once

Our study focused on the usability of setting up 2FA on multiple accounts at once. Ideally, users would not have to go through this process often, but only on rare occasions, such as when a user gets a new phone. The ASQ scores suggest that setting up 2FA for individual
Table 6.1: Pearson correlation coefficient and p-values from comparison of Google ASQ scores between users who experienced technical difficulties during the manual setup of 2FA for Google.

\[
\begin{array}{|c|c|c|c|}
\hline
\text{Pearson correlation coefficient} & \text{ASQ Q1} & \text{ASQ Q2} & \text{ASQ Q3} \\
\hline
\text{p-value} & 0.0457 & 0.0723 & -0.0865 \\
\hline
\end{array}
\]

sites is also easier through an automated process, but the difference in ASQ scores between M-2FA and A-2FA is less pronounced than the difference between the SUS scores for the two systems. This suggests that automated 2FA setup can improve 2FA setup for a single website, but that it provides increased utility for users who are setting up 2FA on multiple accounts at once.

6.4.4 Limitations

Our participants were technically savvy university students already familiar with 2FA and password managers, and so our results may not be applicable to less technical populations. Since we only asked existing password manager users whether they would adopt a password manager 2FA setup tool, our results on the adoptability of our tool are biased. Users who do not use a password manager would be less likely to adopt our tool, since it would also require adoption of a password manager.

We experienced multiple technical difficulties during the study, primarily when attempting to enable the Google prompt during M-2FA. This could have biased participants’ perceptions of the 2FA process on Google, although we observed no statistically significant difference in the ASQ scores of those who experienced technical difficulties and users who did not (see Table 6.1).

We assume that most participants have used 2FA at least occasionally since BYU requires students and faculty to enroll in DUO 2FA in order to access important university online resources. However, we failed to ask participants about their use of 2FA, and so were
unable to analyze how a person’s familiarity with 2FA may have influenced their perceptions about the usability of the M-2FA or A-2FA.
Part IV

Epilogue
Chapter 7

Future Work

Our user study shows that 2FA setup automation is a promising method for improving the usability of 2FA setup. In this chapter, we discuss ideas for future research that could further explore improving the 2FA setup experience.

7.1 Implementation of API

While we hypothesize that implementing our proposed API would greatly improve the usability of 2FA setup through password managers, an implementation of our proposed API on a live website could provide evidence for or against our ideas about the potential benefits of the API. Since we decided to use five popular websites during our user study, the study did not include a website with an implementation of our proposed API. Creating a simple website to implement our API would allow us to measure how quickly 2FA setup might be performed when using our API. A replication of our user study could use API-based setup, with the optional addition of browser-based automation. Based on the participant feedback in our user study, we believe a system using API-based 2FA setup will be rated as even more usable than our automated system was rated.

7.2 Improvement of Proof-of-concept Extension

The feedback from our user study can be used to improve our proof-of-concept extension. The extension can serve as a starting point for real-world password managers to support 2FA setup automation.
7.3 Studying a More Diverse Population

Our population was young, highly-educated, and technically literate compared to the global adult population. Both A-2FA and M-2FA were likely easier for the participants because of their background. We hypothesize that the discrepancy between the usability of M-2FA and A-2FA would be more pronounced among less technically savvy users. These users might experience significantly more difficulty setting up multiple 2FA accounts manually but might not experience the same increased difficulty when using an automated system. Performing a user study with a more diverse population could provide an important perspective on how to encourage 2FA adoption among older, less educated, or less technically savvy users.
Chapter 8

Conclusion

Two-factor authentication provides strong protection against remote account compromise. Despite this, 2FA adoption remains low, in part due to perceptions about 2FA’s low usability. Our work demonstrates the strong potential for using password managers to improve the usability of the 2FA setup process. Our main contributions include:

1. **A first look at the potential for using password managers as a platform for setting up 2FA** — No previous work has explored using password managers as a means for 2FA management.

2. **The first user study exploring the usability of an automated 2FA setup system** — Our 30-person within-subjects user study provides evidence that users may find 2FA setup automation significantly more usable than the current process of manually setting up 2FA.

3. **A proposed API for 2FA setup** — Users found the long wait times during the automated setup process to be off-putting. We propose a web API based on our meta-analysis of the most common forms of 2FA in use today, including U2F, SMS, TOTP, backup codes, and push notifications. This API has the potential to speed up the setup process by an order of magnitude, however, its implementation requires cooperation from supporting websites.

Our work suggests that password managers may be used to significantly improve the usability of 2FA setup. We believe that password managers offering 2FA automation as a
feature may improve 2FA adoption and will improve the scalability of the setup process, allowing users to more easily set up 2FA on multiple accounts. We look to future work to further explore how password managers may be used most effectively to automate 2FA setup.
References


[22] Nikolaos Karapanos, Claudio Marforio, Claudio Soriente, and Srdjan Capkun. Sound-proof: Usable two-factor authentication based on ambient


Appendix A

Materials for User Study

A.1 Qualtrics Exit Survey

Question 1. Please select the range that includes your age.

- 18-24
- 25-34
- 35-44
- 45-54
- 55-64
- 65+

Question 2. Please select your highest level of education.

- Some High School
- High School Diploma or Equivalent
- Some College, No Degree
- Associate Degree
- Bachelor Degree
- Masters Degree
- Professional Degree
- Doctorate Degree

Question 3. How computer savvy do you consider yourself?

- Far above average
- Somewhat above average
- Average
- Somewhat below average
- Far below average

**Question 4.** Which password manager(s) do you use?

- 1Password
- iCloud Keychain
- Bitwarden
- KeePass
- Chrome browser
- LastPass
- Dashlane
- NordPass
- Firefox browser
- Other *(free response)*

**Note:** All participants answered questions about the automated and manual systems, but participants who used the automated system first received the questions about the automated system first, and vice versa.

**Informational text:** In this section, you will answer questions about the (automated)/(manual) setup process. Please provide your thoughts about the system of (automated)/(manual) setup as a whole rather than the experience of setting up individual websites.

**Question 5/9.** What did you like most about the experience of setting up 2FA using the (automated)/(manual) process? *Free response*

**Question 6/10.** What did you like least about the experience of setting up 2FA using the (automated)/(manual) process? *Free response*

**Question 7/11.** What would you do to improve the experience of using the (automated)/(manual) setup process? *Free response*

**Question 8/12.** In the questions below, “system” refers to the (automated setup process)/(manual setup process as a whole). Please rate your level of agreement with each of the
Participants rated each of the following statements from 1 to 5, with 1 being “Strongly disagree” and 5 being “Strongly agree”.

- I think that I would like to use this system frequently.
- I found the system unnecessarily complex.
- I thought the system was easy to use.
- I think that I would need the support of a technical person to be able to use this system.
- I found the various functions in this system were well integrated.
- I thought there was too much inconsistency in this system.
- I would imagine that most people would learn to use this system very quickly.
- I found the system very cumbersome / awkward to use.
- I felt very confident using the system.
- I needed to learn a lot of things before I could get going with this system.

*Informational text:* For the final section, please rate your level of agreement with the following statements.

**Question 13.** I trust the password manager to set up two-factor authentication securely.

- Strongly agree
- Agree
- Somewhat agree
- Neither agree nor disagree
- Somewhat disagree
- Disagree
- Strongly disagree

**Question 14.** I am worried about potential security risks from using a password manager to manage two-factor authentication.

- Strongly agree
- Agree
- Somewhat agree
- Neither agree nor disagree
• Somewhat disagree
• Disagree
• Strongly disagree

**Question 15.** I am more likely to set up two-factor authentication on a personal account after doing this exercise.

• Strongly agree
• Agree
• Somewhat agree
• Neither agree nor disagree
• Somewhat disagree
• Disagree
• Strongly disagree

**Question 16.** I would use an automated two-factor authentication setup system if a password manager I used offered one.

• Strongly agree
• Agree
• Somewhat agree
• Neither agree nor disagree
• Somewhat disagree
• Disagree
• Strongly disagree

### A.2 The After-Scenario Questionnaire

Participants completed the ASQ ten times, twice for each of the five websites. Each question below was rated from 1 to 7, with 1 being “Strongly agree” and 7 being “Strongly disagree”.

1. Overall, I am satisfied with the ease of completing the tasks in this scenario.

2. Overall, I am satisfied with the amount of time it took to complete the tasks in this scenario.

3. Overall, I am satisfied with the support information (online-line help, messages, documentation) when completing the tasks.
A.3 Recruitment flyer

Figure A.1: Flyers used to recruit participants for within-subjects user study.

**Password Manager User Study**

We are conducting research on the usability of an automated two-factor authentication system. The purpose of the study is to find ways to improve the usability of two-factor authentication.

We are looking for participants that meet the following requirements:
- Experience using a standalone password manager (not a browser-based password manager)

Important details about this study include:
- The study will last for 45 minutes and requires only one visit.
- Compensation will be $15.

Find out more and sign up at [byu2fa.youcanbook.me](byu2fa.youcanbook.me)

Internet Security Research Lab
2236 TMCB
Provo, UT 84602-6576
(801) 422-7893

Kent Seamons
Phone: (801) 422-3722
Email: seamons@cs.byu.edu
A.4 Swagger YAML API description

```yaml
swagger: '2.0'
info:
  description: 'Use this API to enroll in 2FA at supported websites.'
  version: 1.0.2
  title: 2FA Enrollment
  host: mywebsite.com
  basePath: /2fa
  schemes:
    - 'https'
paths:
  '/supportedMethods':
    get:
      tags:
        - 2FA
      summary: Lists all supported 2FA methods
      description: Lists all supported 2FA methods
      produces:
        - application/json
      responses:
        '200':
          description: successful operation
          schema:
            type: array
            items:
              $ref: '#/definitions/supportedMethod'
        '400':
          description: Bad Request
          '500':
            description: Internal Server Error
security:
  - petstore_auth:
```

60
'/enabledMethods':

get:

tags:

- 2FA

summary: Lists all 2FA methods currently enabled by the user
description: Lists all enabled 2FA methods

produces:

- application/json

responses:

'200':

description: successful operation

schema:

  type: array

  items:

    type: string

    example: [sms]

'400':

description: Invalid status valu

'500':

description: Internal Server Error

security:

- petstore_auth:

  - 'view:2fa'

  - 'modify:2fa'

'/requestSetup':

post:


tags:

- 2FA

summary: Request enrollment for a 2FA method
description: '

consumes:

- application/json

produces:

- application/json

parameters:

- name: request-object

  in: body

  description: Requested method and with additional information for some methods

  required: true

  schema:
$ref: '#/definitions/RequestObject'

responses:

'200':
  description: OK

'400':
  description: Bad Request

'500':
  description: Internal Server Error

security:
  - petstore_auth:
    - 'modify:2fa'

'/response':

post:
  tags:
    - 2FA
  summary: Return response to challenge
  description: '
  consumes:
    - application/json
  produces:
    - application/json
  parameters:
    - name: request-object
      in: body
      description: Response to requested challenge
      required: true
      schema:
        $ref: '#/definitions/ResponseObject'

responses:

'200':
  description: OK

'400':
  description: Bad Request

'500':
  description: Internal Server Error

security:
  - petstore_auth:
    - 'modify:2fa'

'/remove':

delete:
  tags:
    - 2FA
  summary: Unregisters a second factor
description: ''
produces:
  - application/json
parameters:
  - name: api_key
    in: header
    required: false
    type: string
  - name: petId
    in: path
    description: Pet id to delete
    required: true
    type: integer
    format: int64
responses:
  '200':
    description: OK
  '400':
    description: Bad Request
  '500':
    description: Internal Server Error
security:
  - petstore_auth:
    - 'modify:2fa'
securityDefinitions:
  api_key:
    type: apiKey
    name: api_key
    in: header
  petstore_auth:
    type: oauth2
    authorizationUrl: 'https://mywebsite.com/oauth/authorize'
    flow: implicit
    scopes:
      'view:2fa': View enabled second factors
      'modify:2fa': Add or remove second factors. May require reauthentication

definitions:
  RequestObject:
    type: object
    properties:
      method:
        type: string
factorSpecificInfo:
  type: object
  ChallengeObject:
    type: object
    properties:
      challengeInstruction:
        type: string
  factorSpecificInfo:
    type: object
  ResponseObject:
    type: object
    smsRequest:
      type: object
      properties:
        phoneNumber:
          type: string
          example: 123-456-7890
    smsResponse:
      type: object
      properties:
        code:
          type: string
          example: 123456
    U2FChallenge:
      type: object
      properties:
        appID:
          type: string
        challenge:
          type: string
    U2FResponse:
      type: object
      properties:
        challenge:
          type: string
        KeyHandle:
          type: string
        PublicKey:
          type: string
        AttestationCert:
          type: string
        signature:
          type: string
TOTPChallenge:
  type: object
  properties:
    key:
      type: string
      example: ec9c51d3b6649039c2afdb39118ad56

TOTPResponse:
  type: object
  properties:
    code:
      type: string
      example: 123456
    pushRequest:
      type: object
      properties:
        provider:
          type: string
        providerData:
          type: object
          example:
            phoneNumber: 123-456-6789
            email: "bob@bob.bob"
    codesChallenge:
      type: array
      items:
        type: string
        example: 1234567890
    supportedMethod:
      type: object
      properties:
        methodName:
          type: string
        documentationURL:
          type: string